IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Applicant:	Chen, et al.	Patent Application	
Application No.:	10/789,732	Group Art Unit:	2183
Filed:	February 27, 2004	Examiner:	Faherty, Corey S.

For: PROCESSOR SYNCHRONIZATION IN A MULTI-PROCESSOR COMPUTER SYSTEM

APPEAL BRIEF

Table of Contents

I. Real Party in Interest

The assignee of the present invention is Hewlett-Packard Development Company, L.P.

II. Related Appeals and Interferences

There are no related appeals or interferences known to the Appellants.

III. Status of Claims

Claims 1-5, 7-16 and 18-29 stand rejected. Claims 6 and 17 are cancelled. This appeal involves Claims 1-5, 7-16 and 18-29.

IV. Status of Amendments

All proposed amendments have been entered. An amendment subsequent to the Final Action has not been filed.

V. Summary of Claimed Subject Matter

Independent Claims 1, 12 and 23 of the instant application pertain to embodiments of the present invention.

As recited in Claim 1, "[a] method of synchronizing a plurality of processors of a multi-processor computer system" is described. This embodiment is depicted at least in Fig. 2B. As shown in Fig. 2B, step 202, a plurality of processors to be synchronized are placed in a circular link list, e.g., processors Po, P1, P2, Pn-1 and Pn of Figure 2A. "In step 202, the processors to be synchronized are put on a circular reference arrangement, such as the circular linked list in Fig. 2A" (paragraph 0024, second sentence). "One of the processors is designated the lead processor (e.g., Po) and one of the processors is designated the tail processor (e.g., Pn)" (paragraph 0023, fifth sentence). "In step 204, each processor to be synchronized, except the lead processor, enters its own entry holding loop upon reaching the synchronization point" (paragraph 0024, third sentence). Each processor for synchronization cascadingly leaves its entry holding loop upon the lead processor reaching the synchronization point. "In step 206, each processor to be synchronized is triggered by its immediately preceding processor in a cascading manner to leave its entry holding loop, starting with the triggering by the lead processor for its immediately succeeding processor to leave its entry holding loop" (paragraph 0025, first sentence). "The cascade triggering continues until the tail processor leaves its entry holding loop. Once a processor leaves its entry holding loop, it enters its own exit holding loop" (paragraph 0027). "In step 208, each processor to be synchronized it triggered by its immediately preceding processor in a cascading manner to leave it exit holding loop, starting with the triggering by the tail processor for the lead processor to leave its exit holding loop. ... Note that the tail processor

triggers to the lead processor to leave the exit holding loop after the tail processor reaches the synchronization point and has been triggered to leave its own entry holding loop by its immediately preceding processor" (paragraph 0028, first and fourth sentences).

As recited in Claim 12, "[a] method of synchronization on a synchronization point" is described. This embodiment is depicted at least in Figure 3. "Prior to synchronization, the processors to be synchronized are put on a circular reference arrangement in the manner discussed earlier. The entry holding loop for all processors, except the lead processor Po, is shown in box 302. The exit holding loop [for] all processors is shown in box 304" (paragraph 0031). "In Fig. 3, each of the processors to be synchronized, except the lead processor Po, is put into its own entry holding loop after reaching the synchronization point. ... When the lead processor Po reaches its synchronization point, it triggers (block Po-C) via arrow 306 for its immediately succeeding processor P1 to leave its entry holding loop (p1-B)" (paragraph 0032, first and fifth sentences). "Lead processor Po then enters its exit holding loop Po-D. Once processor P1 leaves its entry holding loop P1-B, it in turn triggers (block P1-C) for its immediately succeeding processor to leave the immediately succeeding processor's entry holding loop. Processor P1 then enters its own exit holding loop P1-D" (paragraph 0033). "The triggering cascades in sequence until the tail processor Pn is triggered (via arrow 308) by its immediately preceding processor Pn-1 to leave its own entry holding loop Pn-B. Note that a processor, except for lead processor Po, does not leave its own entry holding loop until all of its preceding processors have arrived at the synchronization point and have left their own entry holding loops. Also note the each processor enters its own exit holding loop after it has sent the triggering signal to its succeeding processor to leave the succeeding processor's entry holding loop" (paragraph

0034) "In the exit holding loop, a processor continues to check whether it has been triggered (by checking the interrupt request status register, for example) by its immediately preceding processor. Once the tail processor Pn is triggered to leave its entry holding loop Pn-B, it sends a triggering signal to lead processor Po, (via arrow 310) to trigger lead processor Po to leave its exit holding loop Po-D. Upon exiting, lead processor Po in turn triggers (via arrow 312) its immediately succeeding processor P1 to leave its exit holding loop P1-D. The sequence of triggering cascades until the tail processor Pn is triggered (via arrow 314) by its immediately preceding processor Pn-1 to leave it own exit holding loop Pn-D. The exit from the holding loop represents the end of the synchronization sequence for a processor. This is represented in Fig. 3 by states Po-End, Pn-1 End, and Pn-End" (paragraph 0035).

As recited in Claim 23, "[a] method for synchronizing a plurality of processors of a multi-processor computer system on a synchronization point in instantiations of a computer program" is described. This embodiment is depicted at least in Figure 4. "Prior to synchronization, the processors to be synchronized are put on a circular reference arrangement in the manner discussed earlier. In step 402, synchronization starts for a processor when the processor encounters the synchronization point" (paragraph 0036, second and third sentences). "In step 404, the processor inquired whether it is the lead processor Po. If it is not, this processor ascertains in step 406 whether its immediately preceding processor has sent a triggering signal to it. Note that the immediately preceding processor sends a triggering signal after it and all of its preceding processors have reached the synchronization point and after the immediately preceding processor has been triggered by that immediately preceding processor's immediately preceding processor" (paragraph 0037). "If it is ascertained in step 406 that this processor has not been sent a triggering signal by its

immediately preceding processor, the processor stays in the loop (as represented by arrow 407 returning to step 406). On the other hand, if it is ascertained in step 406 that this processor has been sent a triggering signal by its immediately preceding processor, the processor leave the entry holding loop and proceeds to step 408" (paragraph 0039). "In step 408, the processor ascertains whether it is the tail processor Pn. If it is not the tail processor Pn, the processor proceeds to step 410 to send a triggering signal (e.g., using the external interrupt request or EIR mechanism in an embodiment) to its immediately succeeding processor to enable the immediately succeeding processor to leave its own entry holding loop. Note that is the processor is the lead processor Po (as ascertained in step 404), that processor proceeds to directly to step 410" (paragraph 0039). "Thereafter, the processor proceeds to step 412 wherein an inquiry is made as to whether the processor has been sent a triggering signal by its immediately preceding processor to allow it to leave the exit holding loop. If the triggering signal has not been received (as ascertained in step 412), the processor enters an exit holding loop as shown by arrow 413. On the other hand, if the triggering signal has been received from its immediately preceding processor to allow it leave the exit holding loop, the processor proceeds to step 414 wherein it in turn triggers for its immediately succeeding processor to leave its exit holding loop. Thereafter, the synchronization routine ends for the processor (420)" (paragraph 0040). "On the other hand, if it is ascertained in step 408 that it is the tail processor Pn, that tail processor Pn proceeds to step 416 to send a triggering signal (e.g., using the external interrupt request or EIR mechanism in an embodiment) to its immediately succeeding processor to enable the immediately succeeding processor to leave its own exit holding loop. In this case, the triggering signal is sent to the lead processor Po to enable the lead processor Po to leave its own exit holding loop. Thereafter, the tail processor Pn proceeds to step 418 wherein an

inquiry is made as to whether the tail processor Pn has been sent a triggering signal by its immediately preceding processor Pn-1 to allow it to leave the exit holding loop" (paragraph 0041). "If the triggering signal has not been received (as ascertained in step 418), the tail processor Pn enters an exit holding loop as shown by arrow 419. On the other hand, if the triggering signal has been received from its immediately preceding processor Pn-1 to allow tail processor Pn to leave the exit holding loop, tail processor Pn proceeds to step 420 wherein the synchronization routine ends for the tail processor Pn" (paragraph 0042).

VI. Grounds of Rejection to Be Reviewed on Appeal

1. Claims 1-5, 7-16 and 18-29 are rejected under 35 U.S.C. § 102(b) as being anticipated by Johnson et al. (*Cyclical Cascade Chains: A Dynamic Barrier Synchronization Mechanism for Multiprocessor Systems*), referenced from here forward as Johnson.

VII. Argument

1. Whether Claims 1-7 and 9-14 are anticipated under 35 U.S.C. § 102(b) by Johnson.

According to the Final Office Action mailed February 1, 2008, Claims 1-5, 7-16 and 18-29 are rejected under 35 U.S.C. §102(b) as being anticipated by Johnson. Appellants have reviewed Johnson and respectfully submit that the embodiments as recited in Claims 1-5, 7-16 and 18-29 are not anticipated by Johnson for at least the following rationale.

MPEP §2131 provides:

> "A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). ... "The identical invention must be shown in as complete detail as is contained in the ... claim." *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). The elements must be arranged as required by the claim.

Appellants respectfully submit that the rejection of the Claims is improper as the rejection of Claims 1-5, 7-16 and 18-29 does not satisfy the requirements of a *prima facie* case of anticipation as claim embodiments are not met by Johnson. Appellants respectfully submit that Johnson does not teach or suggest each element of the claimed embodiments in the manner set forth in independent Claims 1, 12 and 23.

A. Independent Claim 1

Independent Claim 1 recites (emphasis added):

> A method of synchronizing a plurality of processors of a multi-processor computer system on a synchronization point, said method comprising:
> triggering a first set of processors to enter an entry holding loop in response to said first set of processors encountering said synchronization point before a lead processor associated with said plurality of processors;

triggering said first set of processors to enter an exit holding loop in response to said lead processor encountering said synchronization point; and

triggering said plurality of processors to leave said exit holding loop in response to a tail processor associated with said plurality of processors encountering said synchronization point.

Appellants respectfully submit that Johnson fails to disclose each and every element of Claim 1, arranged as in the claim. Appellants understand Johnson to teach "a dynamic barrier synchronization mechanism with minimal overhead using "cyclical cascade chains." (Abstract). Appellants further understand Johnson to teach "Once all of the dependent processors reach the barrier, the barrier is said to be complete and the processors may continue executing until they reach the next barrier" (page 1, right column, last sentence, first paragraph). Appellants also understand Johnson to teach "Whenever a processor reaches the barrier point, it sets its input signal high. The processor then waits for the *Ready* signal to go high, indicating that all the processors have reached the barrier. The processor then proceeds executing. For the next barrier, a processor reaching the barrier sets its input signal low. The processor then waits for Ready to go low, indicating that all of the processors have reached the barrier. The pattern continues for each successive barrier" (emphasis added; page 2, right column, first paragraph, third to eighth sentences). Appellants do not understand Vahalia to anticipate the claimed embodiment as recited in Claim 1.

Appellants understand Johnson to disclose a plurality of processors, P0-P3, that are in the same processor group. In particular, with reference to page 4, left column, second paragraph, third and fifth sentences, Johnson recites "Taken together, these statements imply that the processors are all in the same group. ... Then each processor defines its location in

the group by the number of the processor one position clockwise from it and by the number of the processor one position counter-clockwise from it, similar to a doubly linked list."

In particular, Appellants respectfully submit that Johnson does not disclose a lead processor nor a tail processor as claimed. In contrast, Appellants understand Johnson to disclose a group of processors in which there is a processor that first reaches the barrier and waits for the other processors. Upon completion of the barrier, as disclosed by Johnson, the processors proceed to the next barrier, at which point, one of the processors will reach the next barrier first. Specifically, as presented above, Appellants submit that Johnson discloses that "Whenever a processor reaches the barrier point, it sets its input signal high. The processor then waits for the *Ready* signal to go high, indicating that all the processors have reached the barrier. The processor then proceeds executing. For the next barrier, a processor reaching the barrier sets its input signal low. The processor then waits for Ready to go low, indicating that all of the processors have reached the barrier. The pattern continues for each successive barrier" (emphasis added; page 2, right column, first paragraph, third to eighth sentences).

In summary, Appellants respectfully submit that the rejections of the Claims are improper as the rejection of Claim 1 does not satisfy the requirements of a *prima facie* case of anticipation as each and every element as set forth in the claim arranged as in the claim are not found in Johnson. In view of Johnson not satisfying the requirements of a *prima facie* case of anticipation, Appellants respectfully submit that independent Claim 1 overcomes the rejection under 35 U.S.C. § 102(b), and that this claim is thus in a condition for allowance.

B. Independent Claims 12 and 23

Appellants respectfully submit that independent Claim 12 recites that an embodiment of the present invention is directed to (emphasis added):

> An article of manufacture comprising a program storage medium having computer readable code embodied therein, wherein executing said computer readable code causes a computer to implement a method of synchronizing a plurality of processors on a synchronization point, said method comprising:
> triggering a first set of processors to enter an entry holding loop in response to said first set of processors encountering said synchronization point before a lead processor associated with said plurality of processors;
> triggering said first set of processors to enter an exit holding loop in response to said lead processor encountering said synchronization point; and
> triggering said plurality of processors to leave said exit holding loop in response to a tail processor encountering said synchronization point.

Independent Claim 23 includes similar embodiments. Claims 2-5 and 7-11 that depend from independent Claim 1, Claims 13-16 and 18-22 that depend from independent Claim 12 and Claims 24-29 the depend from independent Claim 23 also include these embodiments.

Appellants respectfully submit that Johnson fails to disclose "triggering a first set of processors to enter an entry holding loop in response to said first set of processors encountering said synchronization point before a lead processor associated with said plurality of processors; triggering said first set of processors to enter an exit holding loop in response to said lead processor encountering said synchronization point; and triggering said plurality of processors to leave said exit holding loop in response to a tail processor encountering said synchronization point" (emphasis added) as claimed.

Appellants understand Johnson to teach, in a group of processors, and with reference to Single Cyclical Cascade Chain of page 4, left column, "Each processor sends a pulse when it reaches the barrier, instead of raising its input signal and keeping it high until the barrier completes. ... On completion of the barrier, the output signal raises and is used to asynchronously clear the flip-flop. ... The output pulse can be used ... to signal a processor that the barrier has been completed" (page 5, left column, second paragraph). Appellants do not understand Johnson to anticipate the claimed embodiment as recited in Claims 12 and 23.

In particular, Appellants respectfully submit that Johnson does not disclose a lead processor, a tail processor, an entry holding loop, an exit holding loop, or a method for synchronization comprising "triggering a first set of processors to enter an entry holding loop in response to said first set of processors encountering said synchronization point before a lead processor associated with said plurality of processors, triggering said first set of processors to enter an exit holding loop in response to said lead processor encountering said synchronization point; and triggering said plurality of processors to leave said exit holding loop in response to a tail processor encountering said synchronization point" (emphasis added) as claimed. In contrast, as presented above, Appellants understand Johnson to disclose a system in which once all the processors have reached a barrier, the barrier is completed and the processors proceed to the next barrier. Specifically, as presented above, Appellants submit that Johnson discloses that "Whenever a processor reaches the barrier point, it sets its input signal high. The processor then waits for the *Ready* signal to go high, indicating that all the processors have reached the barrier. The processor then proceeds executing. For the next barrier, a processor reaching the barrier sets its input signal low. The processor then waits for Ready to go low, indicating that all of the processors have reached

the barrier. The pattern continues for each successive barrier" (emphasis added; page 2, right column, first paragraph, third to eighth sentences).

In summary, Appellants respectfully submit that the rejections of the Claims are improper as the rejection of Claims 1, 12 and 23 does not satisfy the requirements of a *prima facie* case of anticipation as each and every element as set forth in the claim arranged as in the claim are not found in Johnson.

In view of Johnson not satisfying the requirements of a *prima facie* case of anticipation, Appellants respectfully submit that independent Claims 1, 12 and 23 overcome the rejection under 35 U.S.C. § 102(b), and that these claims are thus in a condition for allowance. Therefore, Appellants respectfully submit that Johnson also does not teach or suggest the additional claimed features as recited in Claims 2-5 and 7-11 that depend from independent Claim 1, as recited in Claims 13-16 and 18-22 that depend from independent Claim 12 and Claims 24-29 that depend from independent Claim 23. Therefore, Appellants respectfully submit that Claims 2-5, 7-11, 13-16, 18-22 and 24-29 also overcome the rejection under 35 U.S.C. § 102(b), and are in a condition for allowance as being dependent on an allowable base claim.

Conclusion

Appellants believe that pending Claims 1-5, 7-16 and 18-29 are not anticipated by Johnson. Therefore, Appellants respectfully submit that the rejections of the Claims are improper as the rejection of Claims 1-5, 7-16 and 18-29 does not satisfy the requirements of a *prima facie* case of anticipation. Accordingly, Appellants respectfully submit that the rejections of Claims 1-5, 7-16 and 18-29 under 35 U.S.C. §102(b) is improper and should be reversed.

The Appellants wish to encourage the Examiner or a member of the Board of Patent Appeals to telephone the Appellants' undersigned representative if it is felt that a telephone conference could expedite prosecution.

Respectfully submitted,
WAGNER BLECHER LLP

Dated: 06/02/2008

/John P. Wagner, Jr./
John P. Wagner, Jr.
Registration No. 35,398
123 Westridge Drive
Watsonville, CA 95076
(408) 377-0500

VIII. Appendix - Clean Copy of Claims on Appeal

1. A method of synchronizing a plurality of processors of a multi-processor computer system on a synchronization point, said method comprising:

triggering a first set of processors to enter an entry holding loop in response to said first set of processors encountering said synchronization point before a lead processor associated with said plurality of processors;

triggering said first set of processors to enter an exit holding loop in response to said lead processor encountering said synchronization point; and

triggering said plurality of processors to leave said exit holding loop in response to a tail processor associated with said plurality of processors encountering said synchronization point.

2. The method of claim 1, further comprising:

creating a circular reference arrangement for said plurality of processors, wherein a first processor from among said plurality of processors is designated said lead processor and a second processor from among said plurality of processors is designated said tail processor, and wherein said lead processor is adjacent to said tail processor in said circular reference arrangement.

3. The method of claim 2 wherein said creating said circular reference arrangement for said plurality of processors, wherein said first processor is designated said lead processor and said second processor is designated said tail processor, and wherein said lead processor is adjacent to said tail processor in said circular reference arrangement, further comprises:

creating said circular reference arrangement for said plurality of processors, wherein said first processor is designated said lead processor and said second processor is designated said tail processor, and wherein said lead processor is adjacent to said tail processor in said circular reference arrangement, said circular reference arrangement representing a circular linked list.

4. The method of claim 2 wherein said triggering said first set of processors to enter said exit holding loop in response to said lead processor encountering said synchronization point further comprises:

triggering said first set of processors to enter said exit holding loop in response to said lead processor encountering said synchronization point, said first set of processors entering said exit holding loop in a cascading manner starting with said lead processor following a sequence established by said circular reference arrangement, with each processor of said first set of processors being triggered by its immediate predecessor in said sequence.

5. The method of claim 4 wherein said triggering said plurality of processors to leave said exit holding loop in response to said tail processor encountering said synchronization point further comprises:

triggering said plurality of processors to leave said exit holding loop in response to said tail processor encountering said synchronization point, said plurality of processors leaving said exit holding loop in a cascading manner starting with said tail processor following said sequence established by said circular reference arrangement, with each processor of said plurality of processors being triggered by its immediate predecessor in said sequence.

7. The method of claim 1, further comprising:

employing a first external interrupt mechanism associated with each of said first set of processors.

8. The method of claim 7 further comprising:

writing to hard physical addresses of each of said first set of processors.

9. The method of claim 7, further comprising:

employing a second external interrupt mechanism associated with each of said plurality of processors.

10. The method of claim 9 further comprising:

writing to hard physical addresses of each of said plurality of processors.

11. The method of claim 1 wherein said triggering said first set of processors to enter said exit holding loop in response to said lead processor encountering said synchronization point further comprises:

triggering said first set of processors to enter said exit holding loop in response to said lead processor encountering said synchronization point utilizing a masked interrupt approach.

12. An article of manufacture comprising a program storage medium having computer readable code embodied therein, wherein executing said computer readable code

causes a computer to implement a method of synchronizing a plurality of processors on a synchronization point, said method comprising:

triggering a first set of processors to enter an entry holding loop in response to said first set of processors encountering said synchronization point before a lead processor associated with said plurality of processors;

triggering said first set of processors to enter an exit holding loop in response to said lead processor encountering said synchronization point; and

triggering said plurality of processors to leave said exit holding loop_in response to a tail processor encountering said synchronization point.

13. The article of manufacture of claim wherein said method further comprises:

creating a circular reference arrangement for said plurality of processors, wherein a first processor from among said plurality of processors is designated said lead processor and a second processor from among said plurality of processors is designated said tail processor, and wherein said lead processor is adjacent to said tail processor in said circular reference arrangement.

14. The article of manufacture of claim 13 wherein said creating said circular reference arrangement for said plurality of processors, wherein said first processor is designated said lead processor and said second processor is designated said tail processor, and wherein said lead processor is adjacent to said tail processor in said circular reference arrangement, further comprises:

creating said circular reference arrangement for said plurality of processors, wherein said first processor is designated said lead processor and said second processor is designated

said tail processor, and wherein said lead processor is adjacent to said tail processor in said circular reference arrangement, said circular reference arrangement representing a circular linked list.

15. The article of manufacture of claim 13 wherein said triggering said first set of processors to enter said exit holding loop in response to said lead processor encountering said synchronization point further comprises:

triggering said first set of processors to enter said exit holding loop in response to said lead processor encountering said synchronization point, said first set of processors entering said exit holding loop in a cascading manner starting with said lead processor following a sequence established by said circular reference arrangement, with each processor of said first set of processors being triggered by its immediate predecessor in said sequence.

16. The article of manufacture of claim 15 wherein said triggering said plurality of processors to leave said exit holding loop in response to said tail processor encountering said synchronization point further comprises:

triggering said plurality of processors to leave said exit holding loop in response to said tail processor encountering said synchronization point, said plurality of processors leaving said exit holding loop in a cascading manner starting with said tail processor following said sequence established by said circular reference arrangement, with each processor of said plurality of processors being triggered by its immediate predecessor in said sequence.

18. The article of manufacture of claim 12 wherein said method further comprises:

employing a first external interrupt mechanism associated with each of said first set of processors.

19. The article of manufacture of claim 18 wherein said method further comprises:

writing to hard physical addresses of each of said first set of processors.

20. The article of manufacture of claim 18 wherein said method further comprises:

employing a second external interrupt mechanism associated with each of said plurality of processors.

21. The article of manufacture of claim 20 wherein said method further comprises:

writing to hard physical addresses of each of said plurality of processors.

22. The article of manufacture of claim 12 wherein said method further comprises:

receiving trigger signals using a masked interrupt approach.

23. A method for synchronizing a plurality of processors of a multi-processor computer system on a synchronization point in instantiations of a computer program, comprising:

implementing a circular reference arrangement for said plurality of processors, each of said plurality of processors having an immediately preceding processor and an immediately succeeding processor, one of said plurality of processors being designated said lead processor, another one of said processors being designated said tail processor, said lead processor immediately succeeding said tail processor in said circular reference arrangement;

keeping a first set of processors in an entry holding loop when said processors of said first set of processors reach said synchronization point, said first set of processors representing said plurality of processors except said lead processor;

cascade triggering along said circular reference arrangement said first set of processors, using a lead processor of said plurality of processors when said lead processor encounters said synchronization point, to enter an exit holding loop, said cascade triggering said first set of processors being performed without accessing a shared memory area of said multi-processor system;

keeping said lead processor in said exit holding loop; and thereafter

cascade triggering along said circular reference arrangement said plurality of processors, using a tail processor of said plurality of processors when said tail processor encounters said synchronization point, to leave said exit holding loop, said cascade triggering said plurality of processors being performed without accessing said shared memory area of said multi-processor system.

24. The method of claim 23 wherein said circular reference arrangement represents a circular linked list.

25. The method of claim 23 wherein said cascade triggering said first set of processors employs a first external interrupt mechanism associated with each of said first set of processors.

26. The method of claim 25 wherein said cascade triggering said first set of processors includes writing to hard physical addresses of each of said first set of processors.

27. The method of claim 25 wherein said cascade triggering said plurality of processors employs a second external interrupt mechanism associated with each of said plurality of processors.

28. The method of claim 27 wherein said cascade triggering said plurality of processors includes writing to hard physical addresses of each of said plurality of processors.

29. The method of claim 23 wherein said cascade triggering employs a masked interrupt approach.

IX. Evidence Appendix

No evidence is herein appended.

X. Related Proceedings Appendix

No related proceedings.